Customer_Churn analysis

Oyelade Oke

2025-02-27

Load necessary libraries and set working directory

Import, load amd explore the data

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

```
# Import and read data file
data <- read.csv("Bank_Churn.csv")

# Explore and review data
str(data)
```

```
## 'data.frame':    10000 obs. of  13 variables:
##  $ CustomerId     : int  15685372 15758813 15803202 15765173 15668309
15679249 15692416 15612494 15779947 15597896 ...
##  $ Surname        : chr  "Azubuike" "Campbell" "Onyekachi" "Lin" ...
##  $ CreditScore    : int  350 350 350 350 350 351 358 359 363 365 ...
##  $ Geography      : chr  "Spain" "Germany" "France" "France" ...
##  $ Gender         : chr  "Male" "Male" "Male" "Female" ...
##  $ Age            : int  54 39 51 60 40 57 52 44 28 30 ...
##  $ Tenure         : int  1 0 10 3 0 4 8 6 6 0 ...
##  $ Balance        : num  152677 109733 0 0 111099 ...
##  $ NumOfProducts  : int  1 2 1 1 1 1 3 1 3 1 ...
##  $ HasCrCard      : int  1 0 1 0 1 1 1 1 1 1 ...
##  $ IsActiveMember : int  1 0 1 0 1 0 0 0 0 0 ...
##  $ EstimatedSalary: num  191973 123602 125824 113796 172321 ...
##  $ Exited         : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
skim(data)
```

Data summary

Name	data
Number of rows	10000
Number of columns	13

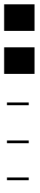
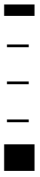
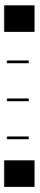
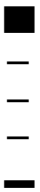
Column type frequency:	
character	3
numeric	10
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Surname	0	1	2	23	0	2932	0
Geography	0	1	5	7	0	3	0
Gender	0	1	4	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
CustomerId	0	1	15690940.57	71936.19	15565701.00	15628528.25	15690738.00	15753233.8	15815690.0	
CreditScore	0	1	650.53	96.65	350.00	584.00	652.00	718.0	850.0	
Age	0	1	38.92	10.49	18.00	32.00	37.00	44.0	92.0	
Tenure	0	1	5.01	2.89	0.00	3.00	5.00	7.0	10.0	
Balance	0	1	76485	623	0.00	0.00	97198	1276	2508	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
			.89	97.41			.54	44.2	98.1	
NumOfProducts	0	1	1.53	0.58	1.00	1.00	1.00	2.0	4.0	
HasCrCard	0	1	0.71	0.46	0.00	0.00	1.00	1.0	1.0	
IsActiveMember	0	1	0.52	0.50	0.00	0.00	1.00	1.0	1.0	
EstimatedSalary	0	1	10009 0.24	575 10.49	11.58	51002 .11	10019 3.91	1493 88.2	1999 92.5	
Exited	0	1	0.20	0.40	0.00	0.00	0.00	0.0	1.0	

```
summary(data)
```

```
##    CustomerId         Surname           CreditScore     Geography
##  Min.   :15565701   Length:10000       Min.   :350.0   Length:10000
##  1st Qu.:15628528   Class :character   1st Qu.:584.0   Class :character
##  Median :15690738   Mode  :character   Median :652.0   Mode  :character
##  Mean   :15690941                      Mean   :650.5
##  3rd Qu.:15753234                      3rd Qu.:718.0
##  Max.   :15815690                      Max.   :850.0
##     Gender                Age             Tenure          Balance
```

```
##  Length:10000       Min.   :18.00   Min.   : 0.000   Min.   :     0
##  Class :character   1st Qu.:32.00   1st Qu.: 3.000   1st Qu.:     0
##  Mode  :character   Median :37.00   Median : 5.000   Median : 97199
##                     Mean   :38.92   Mean   : 5.013   Mean   : 76486
##                     3rd Qu.:44.00   3rd Qu.: 7.000   3rd Qu.:127644
##                     Max.   :92.00   Max.   :10.000   Max.   :250898
##  NumOfProducts    HasCrCard      IsActiveMember   EstimatedSalary
##  Min.   :1.00   Min.   :0.0000   Min.   :0.0000   Min.   :    11.58
##  1st Qu.:1.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 51002.11
##  Median :1.00   Median :1.0000   Median :1.0000   Median :100193.91
##  Mean   :1.53   Mean   :0.7055   Mean   :0.5151   Mean   :100090.24
##  3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:149388.25
##  Max.   :4.00   Max.   :1.0000   Max.   :1.0000   Max.   :199992.48
##      Exited
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2037
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

Data transformation

```r
# Exited variable
data$Exited <- ifelse(data$Exited == 1, "Churned", "Retention")

# Convert back to a factor for proper categorical handling
data$Exited <- factor(data$Exited, levels = c("Retention", "Churned"))

# Verify the changes
table(data$Exited)
```

```
##
## Retention   Churned
##      7963      2037
```

```r
# Transform numeric variables into categorical
# Categorize 'Age' into age groups

# data$AgeGroup <- cut(data$Age, breaks = c(18, 30, 40, 50, 60, 100),
                       #labels = c("18-30", "31-40", "41-50", "51-60", "60+"),
right = FALSE)
data <- data %>%
  mutate(AgeGroup = case_when(
    Age <= 30 ~ "18-30",
    Age > 30 & Age <= 45 ~ "31-45",
    Age > 45 & Age <= 60 ~ "46-60",
    Age > 60 ~ "60+"
  ))
```

```r
# Categorize CreditScore into Low, Medium, and High based on specified ranges
data$CreditScoreCategory <- cut(data$CreditScore,
                                breaks = c(-Inf, 584, 718, Inf),
                                labels = c("Low", "Medium", "High"),
                                right = TRUE)  # Ensure <=584, >584 & <=718,
>718
table(data$CreditScoreCategory)

##
##    Low Medium   High
##   2534   5003   2463

# Categorize 'Balance' into bins
data$BalanceCategory <- cut(data$Balance, breaks = c(-1, 0, 50000, 100000,
200000, max(data$Balance)),
                            labels = c("Zero", "Low", "Medium", "High", "Very
High"), right = FALSE)
table(data$BalanceCategory)

##
##      Zero       Low    Medium      High Very High
##         0      3692      1509      4765        33
```

Statistical Tests: This identifies variables statistical relevant for influencing customer churn

Fit a logistic regression model

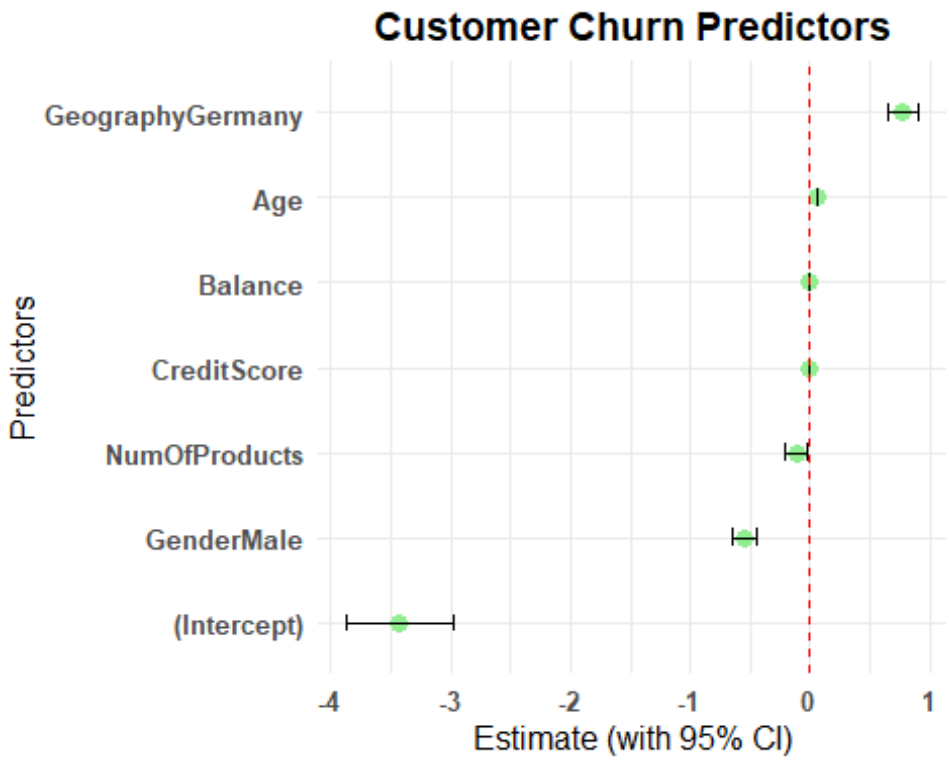
```r
##
## Call:
## glm(formula = Exited ~ Age + Gender + Geography + CreditScore +
##     Balance + NumOfProducts, family = binomial, data = data)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.428e+00  2.257e-01 -15.189  < 2e-16 ***
## Age               6.338e-02  2.422e-03  26.168  < 2e-16 ***
## GenderMale       -5.434e-01  5.332e-02 -10.190  < 2e-16 ***
## GeographyGermany  7.756e-01  6.607e-02  11.739  < 2e-16 ***
## GeographySpain    2.353e-02  6.957e-02   0.338  0.73516
## CreditScore      -8.032e-04  2.741e-04  -2.930  0.00339 **
## Balance           2.579e-06  5.072e-07   5.086 3.66e-07 ***
## NumOfProducts    -1.169e-01  4.637e-02  -2.521  0.01170 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10109.8  on 9999  degrees of freedom
## Residual deviance:  8937.2  on 9992  degrees of freedom
## AIC: 8953.2
```

```
##
## Number of Fisher Scoring iterations: 4

##                       Estimate   Std. Error    z value      Pr(>|z|)
## (Intercept)      -3.427866e+00 2.256852e-01 -15.188706  4.201303e-52
## Age               6.337692e-02 2.421923e-03  26.168013 6.147868e-151
## GenderMale       -5.433602e-01 5.332443e-02 -10.189704  2.204340e-24
## GeographyGermany  7.755998e-01 6.606969e-02  11.739117  8.032061e-32
## CreditScore      -8.031645e-04 2.741393e-04  -2.929768  3.392153e-03
## Balance           2.579369e-06 5.071571e-07   5.085936  3.658165e-07
## NumOfProducts    -1.168980e-01 4.636990e-02  -2.520989  1.170256e-02

## [1] "matrix" "array"
```

Visualisations

Forest Plot of significant variables, showing their respective coefficients and direction of influence

```
significant_vars <- significant_vars %>%
  mutate(
    Predictor = rownames(significant_vars),  # Add row names as a column for
predictors
    LowerCI = Estimate - 1.96 * `Std. Error`,  # Calculate 95% confidence
intervals
    UpperCI = Estimate + 1.96 * `Std. Error`
  )

ggplot(significant_vars, aes(x = Estimate, y = reorder(Predictor, Estimate)))
+
  geom_point(size = 3, color = "lightgreen") +  # Point for estimates
  geom_errorbarh(aes(xmin = LowerCI, xmax = UpperCI), height = 0.2, color =
"black") +  # Horizontal error bars
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +  #
Vertical line at 0
  labs(
    title = "Customer Churn Predictors",
    x = "Estimate (with 95% CI)",
    y = "Predictors"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10, face = "bold")
  )
```

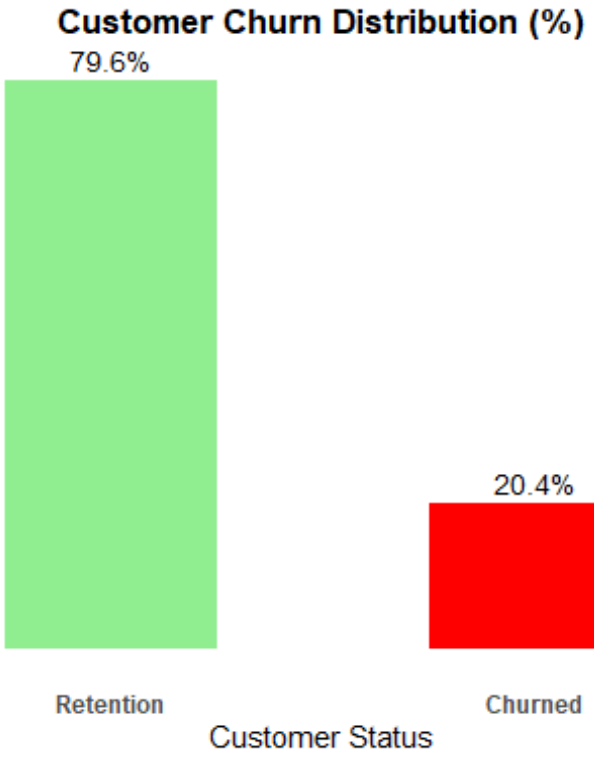


Other Exploratory Visualisations

```
# Customer churn distribution
# Calculate percentages for each category
churn_percent <- data %>%
  group_by(Exited) %>%
  summarise(Percentage = n() / nrow(data) * 100)

# Create churn distribution with percentages and labels
ggplot(data, aes(x = Exited, fill = Exited)) +
  geom_bar(aes(y = (..count..) / sum(..count..) * 100), width = 0.5) +  #
Reduce bar width
  geom_text(data = churn_percent, aes(x = Exited, y = Percentage, label =
sprintf("%.1f%%", Percentage)),
            vjust = -0.5, size = 3.8) +  # Add percentage labels above bars
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +  # Format
y-axis as percentages
  scale_fill_manual(values = c("Retention" = "lightgreen", "Churned" =
"red")) +  # Set colors for Retention and Churned
  labs(title = "Customer Churn Distribution (%)", x = "Customer Status", y =
"Percentage") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),  # Remove gridlines
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold"),  #
Center and style title
    axis.text.y = element_blank(),  # Remove y-axis text
```

```
    axis.ticks.y = element_blank(),  # Remove y-axis ticks
    axis.title.y = element_blank(),  # Remove y-axis title
    axis.text.x = element_text(face = "bold"),
    axis.title.x = element_text(),
    legend.position = "none"  # Remove legend
  )
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2
3.4.0.
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

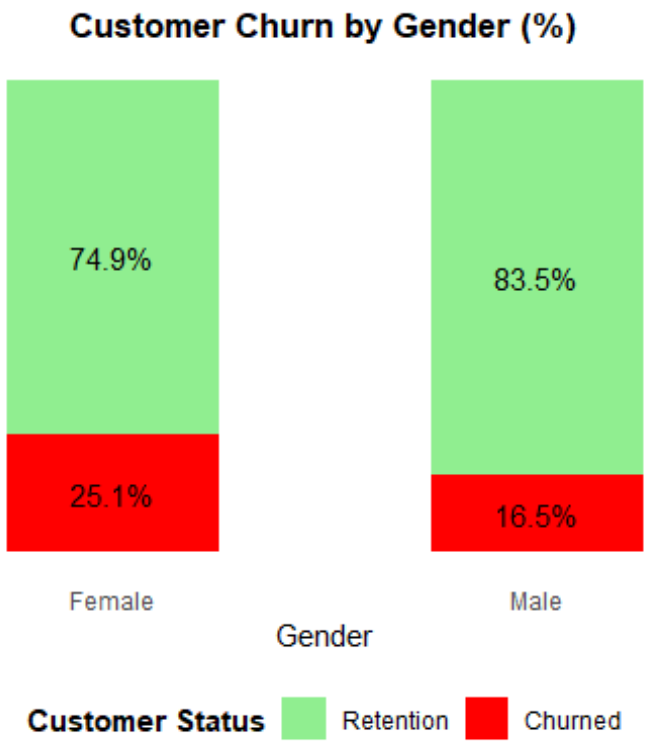



```
# Customer Churn by Gender
# Prepare data for Gender and churn proportions
gender_data <- data %>%
  group_by(Gender, Exited) %>%
  summarise(Count = n()) %>%
  group_by(Gender) %>%
  mutate(Proportion = Count / sum(Count) * 100)
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```

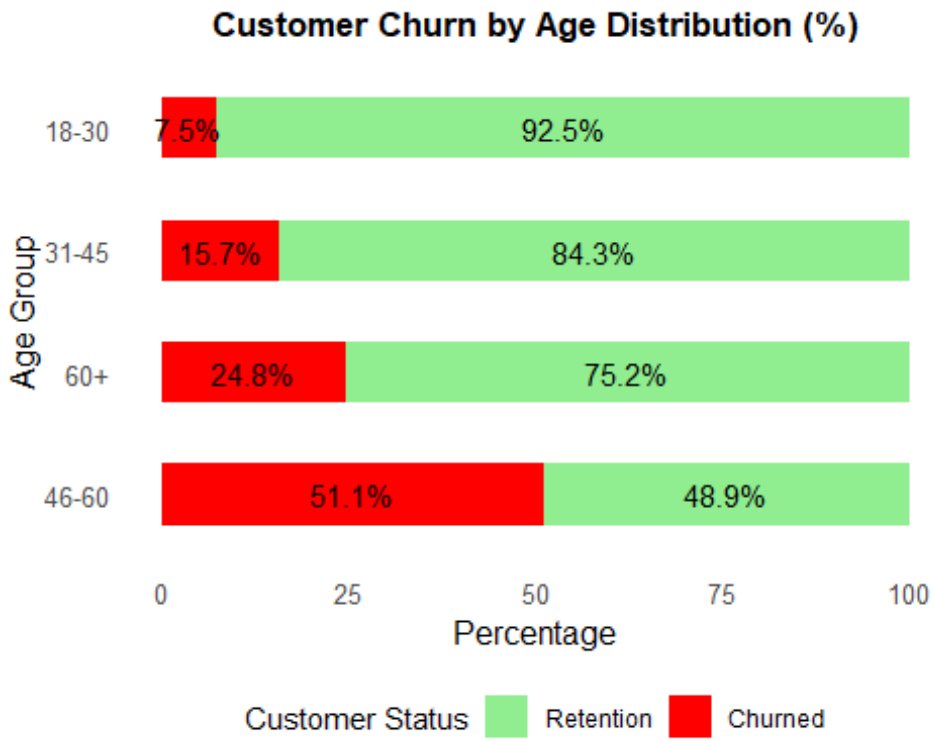
```
# Plot stacked bar chart for Gender with percentage labels
ggplot(gender_data, aes(x = Gender, y = Proportion, fill =
```

```r
as.factor(Exited))) +
  geom_bar(stat = "identity", width = 0.5) +  # Stacked proportional bars without border
  geom_text(aes(label = paste0(round(Proportion, 1), "%")),  # Add percentage labels
            position = position_stack(vjust = 0.5), size = 4, color = "black") +
  scale_fill_manual(values = c("lightgreen", "red"), labels = c("Retention", "Churned")) +
  labs(title = "Customer Churn by Gender (%)",
       x = "Gender", y = NULL, fill = "Customer Status") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold"),
    axis.text.y = element_blank(),       # Remove y-axis values
    axis.ticks.y = element_blank(),      # Remove y-axis ticks
    panel.grid = element_blank(),        # Remove gridlines
    legend.title = element_text(face = "bold"),
    legend.position = "bottom"
  )
```



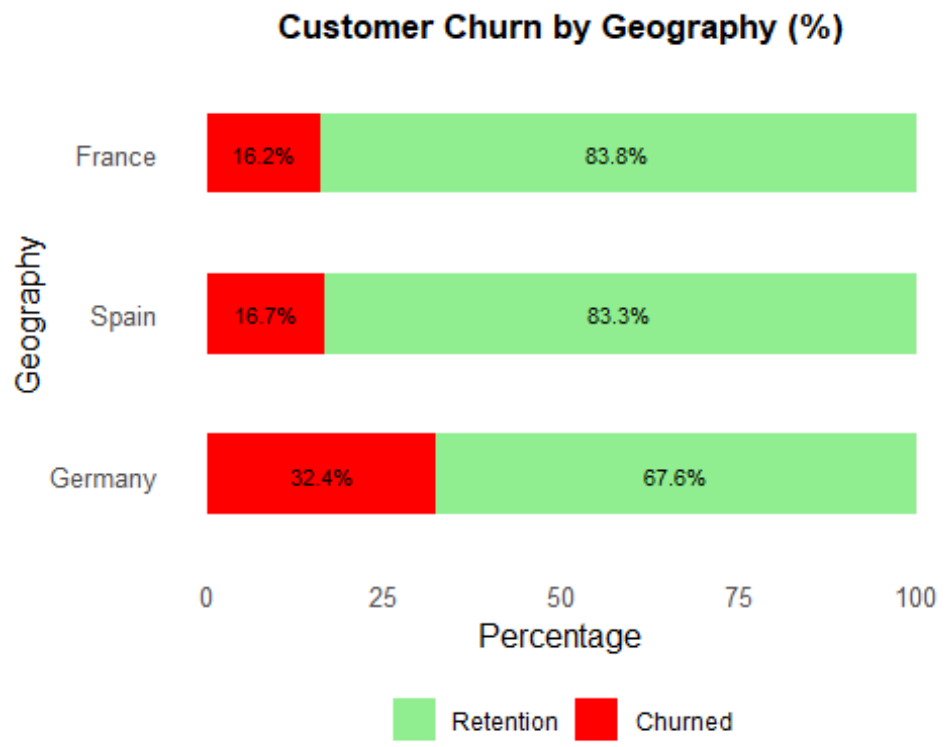

```r
# Customer Churn Rate by Age Group
# Churn by age group
age_churn_summary <- data %>%
  group_by(AgeGroup, Exited) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = Count / sum(Count) * 100)
```

```
## `summarise()` has grouped output by 'AgeGroup'. You can override using the
## `.groups` argument.

# Visualize churn by age group
# Stacked bar chart for churn by age group
# Update chart to flip coordinates and rank churned customers
ggplot(age_churn_summary %>% arrange(desc(Exited), desc(Percentage)),
       aes(x = reorder(AgeGroup, -Percentage *
as.numeric(as.factor(Exited))),
           y = Percentage, fill = as.factor(Exited))) +
  geom_bar(stat = "identity", width = 0.5) +
  coord_flip() +  # Flip the chart
  scale_fill_manual(values = c("lightgreen", "red"), labels = c("Retention",
"Churned")) +
  labs(
    title = "Customer Churn by Age Distribution (%)",
    x = "Age Group",
    y = "Percentage",
    fill = "Customer Status"
  ) +
  geom_text(
    aes(label = sprintf("%.1f%%", Percentage)),
    position = position_stack(vjust = 0.5), size = 4, color = "black"
  ) +  # Add percentage labels inside bars
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold"),  #
Center title
    axis.text = element_text(size = 10),  # Adjust axis text size
    axis.title = element_text(size = 12),  # Adjust axis title size
    legend.position = "bottom",
    panel.grid = element_blank()  # Remove gridlines
  )
```



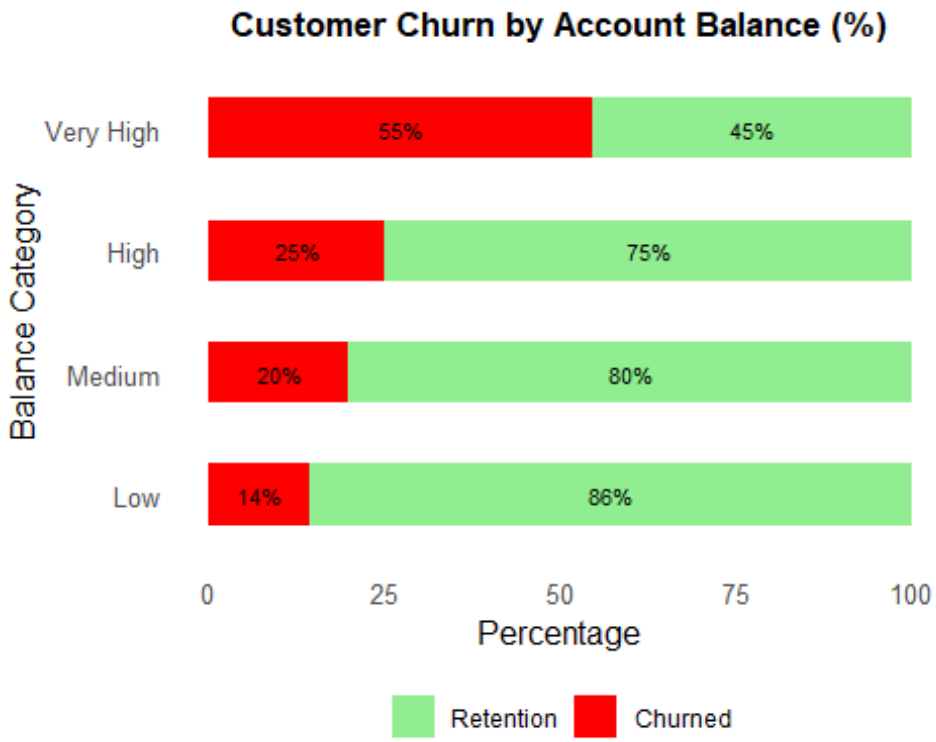

```r
# Customer Churn Rate by Geography

# Calculate percentages for churn by geography
geo_churn_percent <- data %>%
  group_by(Geography, Exited) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = Count / sum(Count) * 100)

## `summarise()` has grouped output by 'Geography'. You can override using
the
## `.groups` argument.

# Create Churn by Geography with facet_wrap
# Flip chart and rank churned customers
ggplot(geo_churn_percent %>% arrange(desc(Percentage)),
       aes(x = reorder(Geography, -Percentage), y = Percentage, fill =
Exited)) +
  geom_bar(stat = "identity", width = 0.5) +  # Horizontal bar chart
  geom_text(aes(label = sprintf("%.1f%%", Percentage)),
            position = position_stack(vjust = 0.5), size = 3, color =
"black") +  # Add percentage labels
  coord_flip() +  # Flip the chart
  scale_fill_manual(values = c("Retention" = "lightgreen", "Churned" =
"red")) +  # Set custom colors
  labs(
    title = "Customer Churn by Geography (%)",
    x = "Geography",
```

```
    y = "Percentage",
    fill = "Customer Status"
  ) +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),  # Remove gridlines
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold"),  #
Center and style title
    axis.text = element_text(size = 10),  # Adjust axis text size
    axis.title = element_text(size = 12),  # Adjust axis title size
    strip.text = element_text(size = 13, face = "bold"),  # Style facet
labels
    legend.title = element_blank(),  # Remove legend title
    panel.border = element_blank(),
    legend.position = "bottom" # Add borders around facets
  )
```



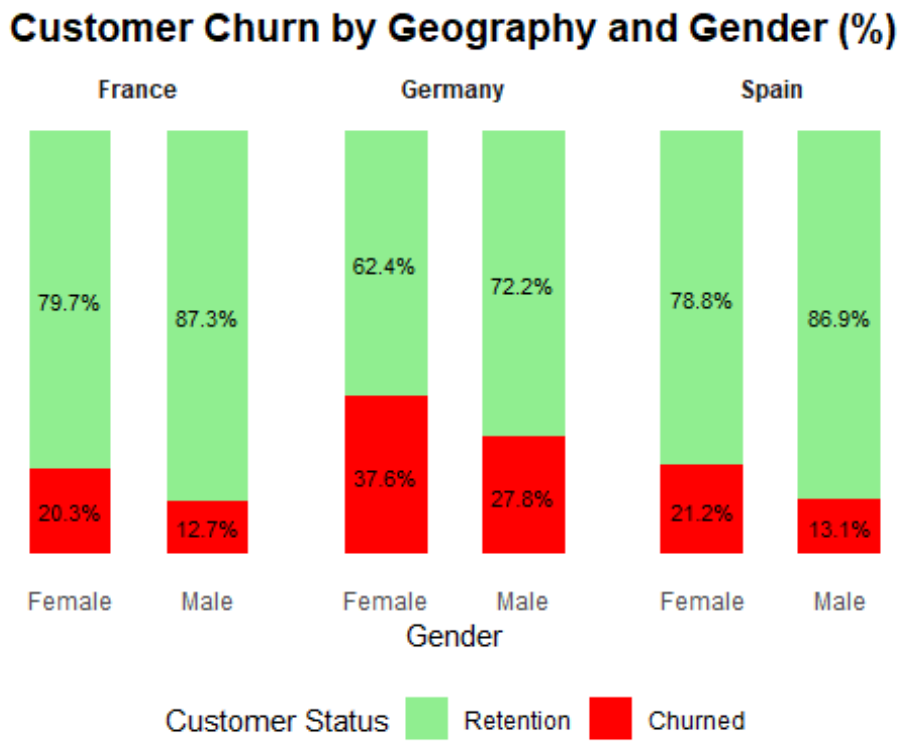

```
# Customer Churn Rate by BalanceCategory
# Prepare data for BalanceCategory and churn proportions
balance_data <- data %>%
  filter(!is.na(BalanceCategory)) %>%
  group_by(BalanceCategory, Exited) %>%
  summarise(Count = n()) %>%
  group_by(BalanceCategory) %>%
  mutate(Percentage = Count / sum(Count) * 100)
```

```
## `summarise()` has grouped output by 'BalanceCategory'. You can override
using
## the `.groups` argument.

# Plot stacked bar chart for BalanceCategory with percentage labels
ggplot(balance_data %>% arrange(desc(Percentage)),
       aes(x = reorder(BalanceCategory, -Percentage), y = Percentage, fill =
Exited)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(label = sprintf("%.f%%", Percentage)),  # Add percentage
labels
            position = position_stack(vjust = 0.5), size = 3, color =
"black") +
  coord_flip() +
  scale_fill_manual(values = c("Retention" = "lightgreen", "Churned" =
"red")) +
  labs(
    title = "Customer Churn by Account Balance (%)",
       x = "Balance Category",
       y = "Percentage",
       fill = "Customer Status") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    plot.title = element_text(hjust = 0.5, size = 13, face = "bold"),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    strip.text = element_text(size = 13, face = "bold"),
    legend.title = element_blank(),
    panel.border = element_blank(),
    legend.position = "bottom"
  )
```




```r
# Customer Churn by Geography and Gender
geo_drivers_data <- data %>%
  filter(!is.na(Geography), !is.na(Gender)) %>%
  group_by(Geography, Gender, Exited) %>%
  summarise(Count = n(), .groups = "drop") %>%
  group_by(Geography, Gender) %>%
  mutate(Proportion = Count / sum(Count) * 100)

# Plot stacked bar chart for Geography with key drivers (e.g Gender)
ggplot(geo_drivers_data, aes(x = Gender, y = Proportion, fill =
as.factor(Exited))) +
  geom_bar(stat = "identity", width = 0.6) +  # Stacked proportional bars
  geom_text(aes(label = paste0(round(Proportion, 1), "%")),  # Add percentage
labels
            position = position_stack(vjust = 0.5), size = 3, color =
"black") +
  scale_fill_manual(values = c("lightgreen", "red"), labels = c("Retention",
"Churned")) +
  facet_wrap(~ Geography) +  # Separate charts for each geography
  labs(
    title = "Customer Churn by Geography and Gender (%)",
    x = "Gender",
    y = NULL,
    fill = "Customer Status"
  ) +
  theme_minimal() +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.text.y = element_blank(),       # Remove y-axis values
    axis.ticks.y = element_blank(),      # Remove y-axis ticks
    panel.grid = element_blank(),        # Remove gridlines
    legend.position = "bottom",
    strip.text = element_text(face = "bold")
  )
```




```
# Customer Churn Balance by Geography
data <- data %>%
  filter(!is.na(Balance)) %>%  # Remove NA values from Balance before
categorization
  mutate(BalanceCategory = cut(Balance,
                               breaks = c(0, 1, 50000, 100000, 200000,
max(Balance, na.rm = TRUE) + 1),  # Add 1 to max
                               labels = c("Zero", "Low", "Medium", "High",
"Very High"),
                               right = FALSE, include.lowest = TRUE))  #
Include 0 and adjust intervals

# Plot stacked bar chart with facets for Geography
ggplot(data, aes(x = BalanceCategory, fill = as.factor(Exited))) +
  geom_bar(position = "fill", width = 0.6) +  # Stacked proportional bars
  facet_wrap(~ Geography) +  # Separate charts for each Geography
  scale_fill_manual(values = c("lightgreen", "red"), labels = c("Retention",
"Churned")) +
```

```r
  scale_y_continuous(labels = percent_format()) +  # Convert y-axis to
percentage
  labs(title = "Customer Churn by Geography and Balance",
       x = "Balance Category", y = "Percentage", fill = "Customer Status") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    strip.text = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.border = element_rect(color = "black", fill = NA)
  )
```